6:20-cv-00350

EXHIBIT B

PRELIMINARY CLAIM CHART (Subject to supplementation and amendment based on acquisition of further information)

Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")
8. An apparatus comprising: a detection circuit configured to generate a signal having on event condition; and	The accused product utilizes an apparatus comprising: a detection circuit (e.g., a battery monitoring circuit) configured to generate a signal (e.g., voltage or current notification) having on event condition (e.g., if state is high or low). As shown below, the LG V30 utilizes a Qualcomm Snapdragon 835 processor.

EXHIBIT B

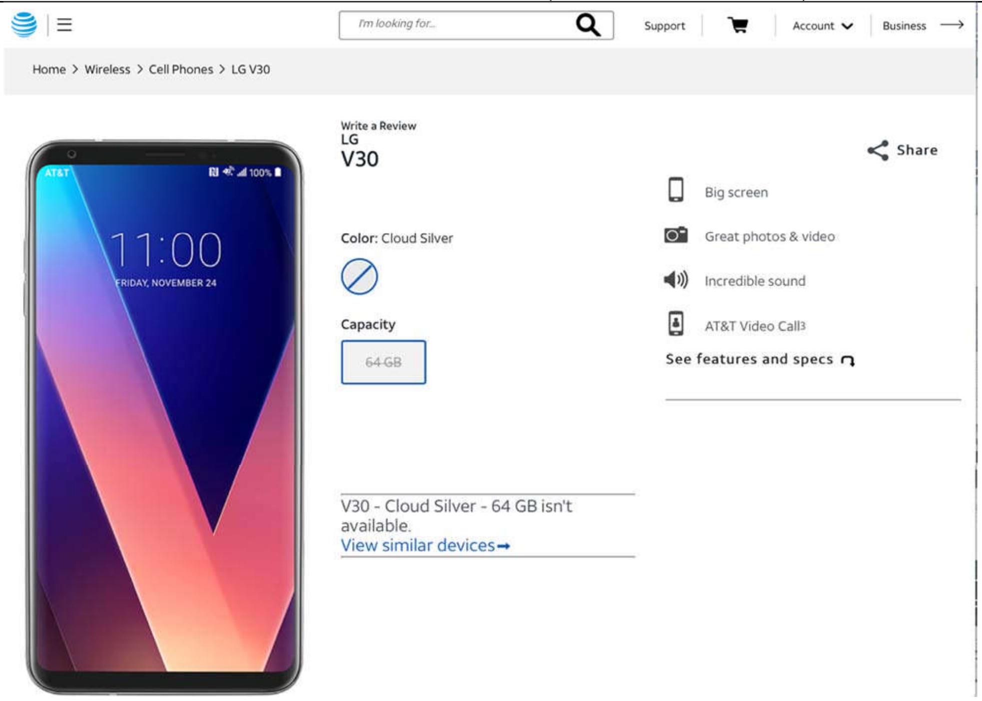
Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")
	 Home > Wireless > Cell Phones > LG V30 Write a Review LG V30 Color: Cloud Silver Capacity ~~64 GB~~ Big screen Great photos & video Incredible sound AT&T Video Call3 See features and specs V30 - Cloud Silver - 64 GB isn't available. View similar devices→ https://www.att.com/buy/phones/lg-v30-64gb-cloud-silver.html

EXHIBIT B

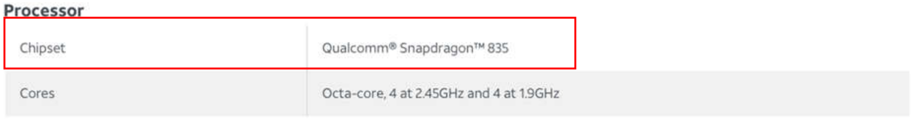
Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")
	 Processor Chipset — Qualcomm® Snapdragon™ 835 Cores — Octa-core, 4 at 2.45GHz and 4 at 1.9GHz https://www.att.com/buy/phones/lg-v30-64gb-cloud-silver.html As shown below, the Snapdragon 835 includes a battery monitoring circuit that generates a signal based upon the occurrence of a certain condition (in this case voltage variances for normal values). https://www.qualcomm.com/products/snapdragon-835-mobile-platform

EXHIBIT B

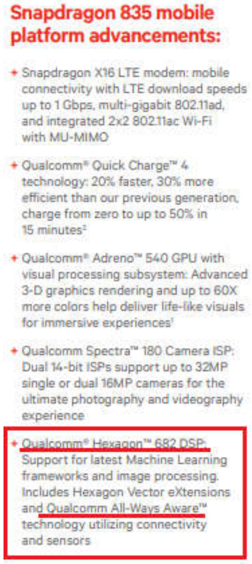
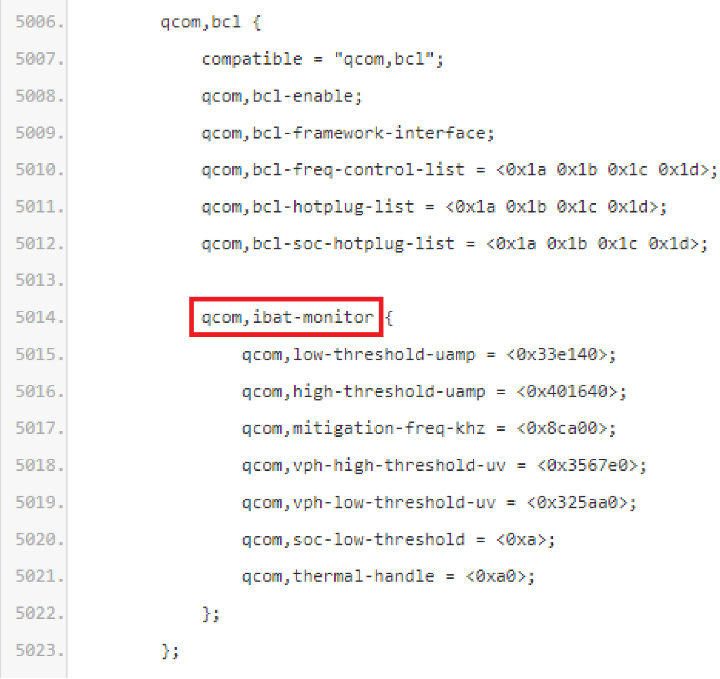
Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")
	 **Snapdragon 835 mobile platform advancements:** + Snapdragon X16 LTE modem: mobile connectivity with LTE download speeds up to 1 Gbps, multi-gigabit 802.11ad, and integrated 2x2 802.11ac Wi-Fi with MU-MIMO + Qualcomm® Quick Charge™ 4 technology: 20% faster, 30% more efficient than our previous generation, charge from zero to up to 50% in 15 minutes[2] + Qualcomm® Adreno™ 540 GPU with visual processing subsystem: Advanced 3-D graphics rendering and up to 60X more colors help deliver life-like visuals for immersive experiences[1] + Qualcomm Spectra™ 180 Camera ISP: Dual 14-bit ISPs support up to 32MP single or dual 16MP cameras for the ultimate photography and videography experience + Qualcomm® Hexagon™ 682 DSP: Support for latest Machine Learning frameworks and image processing. Includes Hexagon Vector eXtensions and Qualcomm All-Ways Aware™ technology utilizing connectivity and sensors https://www.qualcomm.com/media/documents/files/snapdragon-835-mobile-platform-product-brief.pdf

EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T's Sale of the LG V30 ("The Accused Product")</th></tr>
<tr><td></td><td>



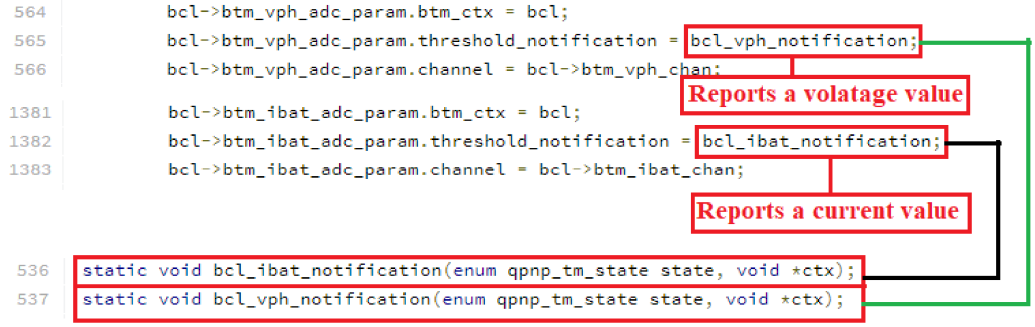
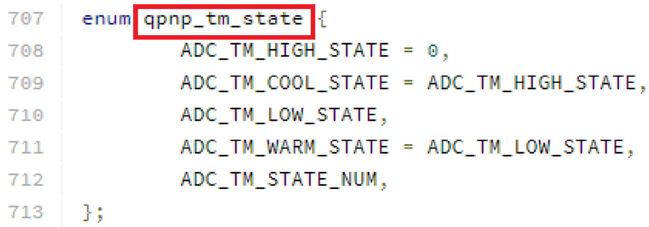
```
qcom,bcl {
    compatible = "qcom,bcl";
    qcom,bcl-enable;
    qcom,bcl-framework-interface;
    qcom,bcl-freq-control-list = <0x1a 0x1b 0x1c 0x1d>;
    qcom,bcl-hotplug-list = <0x1a 0x1b 0x1c 0x1d>;
    qcom,bcl-soc-hotplug-list = <0x1a 0x1b 0x1c 0x1d>;

    qcom,ibat-monitor {
        qcom,low-threshold-uamp = <0x33e140>;
        qcom,high-threshold-uamp = <0x401640>;
        qcom,mitigation-freq-khz = <0x8ca00>;
        qcom,vph-high-threshold-uv = <0x3567e0>;
        qcom,vph-low-threshold-uv = <0x325aa0>;
        qcom,soc-low-threshold = <0xa>;
        qcom,thermal-handle = <0xa0>;
    };
};
```

https://pastebin.com/U0i7nP4P

</td></tr>
</table>

EXHIBIT B

Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")

	
	`bcl->btm_vph_adc_param.btm_ctx = bcl;`
	`bcl->btm_vph_adc_param.threshold_notification = bcl_vph_notification;`
	`bcl->btm_vph_adc_param.channel = bcl->btm_vph_chan;`
	`bcl->btm_ibat_adc_param.btm_ctx = bcl;`
	`bcl->btm_ibat_adc_param.threshold_notification = bcl_ibat_notification;`
	`bcl->btm_ibat_adc_param.channel = bcl->btm_ibat_chan;`
	`static void bcl_ibat_notification(enum qpnp_tm_state state, void *ctx);`
	`static void bcl_vph_notification(enum qpnp_tm_state state, void *ctx);`
	Reports a volatage value
	Reports a current value
	https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-angler-3.10-nougat/drivers/power/battery_current_limit.c
	
	`enum qpnp_tm_state {`
	`        ADC_TM_HIGH_STATE = 0,`
	`        ADC_TM_COOL_STATE = ADC_TM_HIGH_STATE,`
	`        ADC_TM_LOW_STATE,`
	`        ADC_TM_WARM_STATE = ADC_TM_LOW_STATE,`
	`        ADC_TM_STATE_NUM,`
	`};`
	https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-asus-3.10-nougat-mr1-wear-release/include/linux/qpnp/qpnp-adc.h

a storage circuit configured to store said event;	The accused product comprises a storage circuit (e.g., L2 cache) configured to store said event (e.g., if state is high or low). As shown below, the Snapdragon 835 includes an L2 cache that stores voltage variance events.

EXHIBIT B

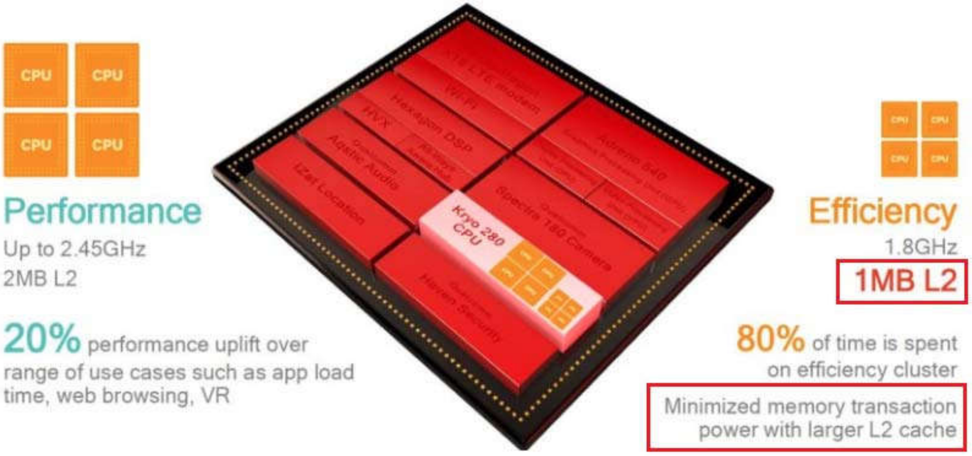
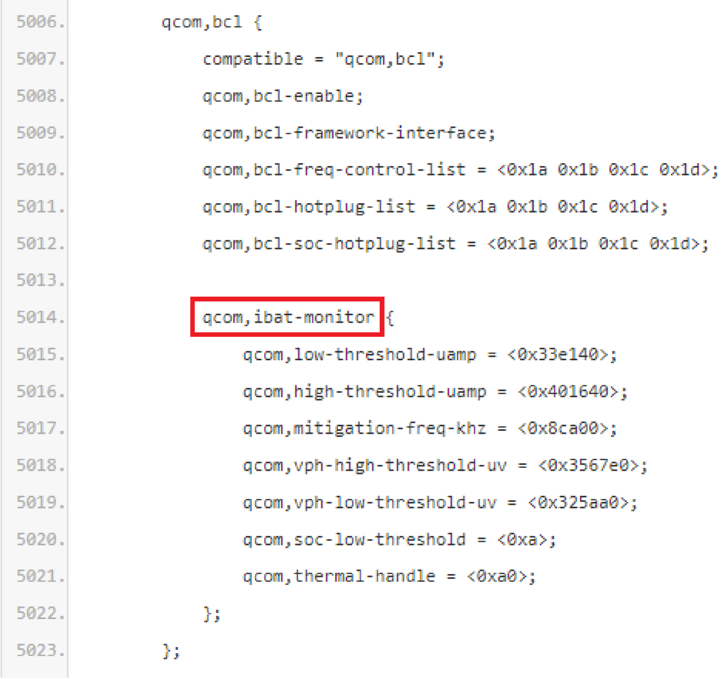
Pat 6,819,539	AT&T’s Sale of the LG V30 (“The Accused Product”)
	 CPU CPU CPU CPU X16 LTE modem Wi-Fi Hexagon DSP HVX Aqstic Audio IZat Location Kryo 280 CPU Spectra 180 Camera Adreno 540 Haven Security Performance Up to 2.45GHz 2MB L2 20% performance uplift over range of use cases such as app load time, web browsing, VR Efficiency 1.8GHz 1MB L2 80% of time is spent on efficiency cluster Minimized memory transaction power with larger L2 cache https://www.androidauthority.com/qualcomm-details-snapdragon-835-735688/
a table configured to store a plurality of event types; and	The accused product comprises a table (e.g., a table containing various thresholds) configured to store a plurality of event types (e.g., if state is high or low). As shown in the code below, the Snapdragon 835 utilizes a table that defines various voltage conditions and their corresponding thresholds.

EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T’s Sale of the LG V30 (“The Accused Product”)</th></tr>
<tr><td></td><td>

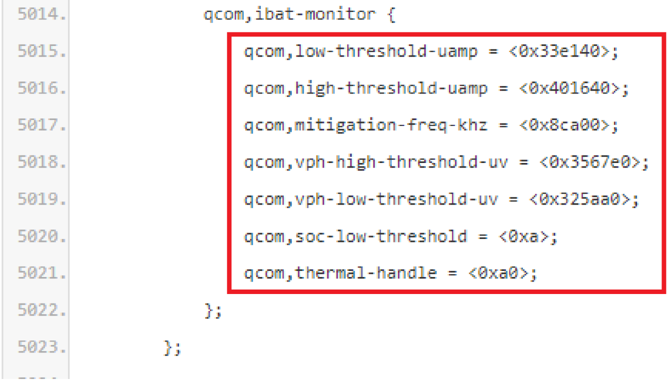


```
qcom,bcl {
    compatible = "qcom,bcl";
    qcom,bcl-enable;
    qcom,bcl-framework-interface;
    qcom,bcl-freq-control-list = <0x1a 0x1b 0x1c 0x1d>;
    qcom,bcl-hotplug-list = <0x1a 0x1b 0x1c 0x1d>;
    qcom,bcl-soc-hotplug-list = <0x1a 0x1b 0x1c 0x1d>;

    qcom,ibat-monitor {
        qcom,low-threshold-uamp = <0x33e140>;
        qcom,high-threshold-uamp = <0x401640>;
        qcom,mitigation-freq-khz = <0x8ca00>;
        qcom,vph-high-threshold-uv = <0x3567e0>;
        qcom,vph-low-threshold-uv = <0x325aa0>;
        qcom,soc-low-threshold = <0xa>;
        qcom,thermal-handle = <0xa0>;
    };
};
```

https://pastebin.com/U0i7nP4P
</td></tr>
</table>

EXHIBIT B

Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")

	
	`qcom,ibat-monitor {`
	`    qcom,low-threshold-uamp = <0x33e140>;`
	`    qcom,high-threshold-uamp = <0x401640>;`
	`    qcom,mitigation-freq-khz = <0x8ca00>;`
	`    qcom,vph-high-threshold-uv = <0x3567e0>;`
	`    qcom,vph-low-threshold-uv = <0x325aa0>;`
	`    qcom,soc-low-threshold = <0xa>;`
	`    qcom,thermal-handle = <0xa0>;`
	`};`
	`};`
	https://pastebin.com/U0i7nP4P

a circuit configured to (i) reset when said event condition is a first predetermined type and (ii) implement recover action when said event condition is a second predetermined type, wherein said first and second predetermined types are determined in response to a comparison of said event to said plurality of event types stored in said table.	The accused product comprises a circuit (e.g., resource power manager circuit) configured to (i) reset (e.g., cpu_down) when said event condition is a first predetermined type (e.g., when bcl_soc_state == BCL_LOW_THRESHOLD OR bcl_vph_state == BCL_LOW_THRESHOLD) and (ii) implement recover action (e.g., cpu_up) when said event condition is a second predetermined type (e.g., when bcl_soc_state is not equal to BCL_LOW_THRESHOLD, bcl_vph_state is not equal to BCL_LOW_THRESHOLD and bcl_ibat_state is not equal to BCL_HIGH_THRESHOLD), wherein said first and second predetermined types are determined in response to a comparison of said event to said plurality of event types stored in said table (e.g. the comparison of collected values with stored thresholds).

EXHIBIT B

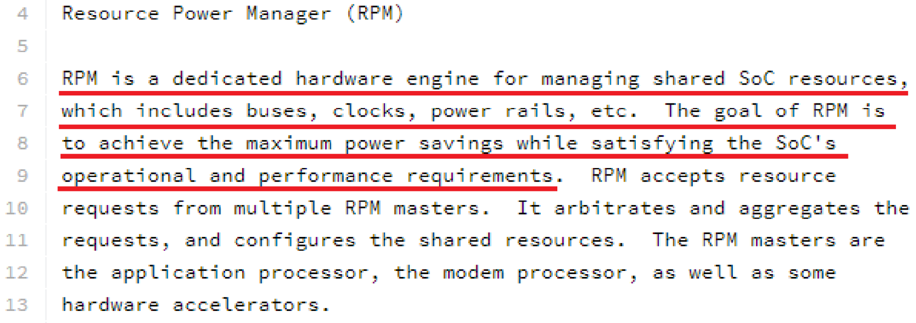
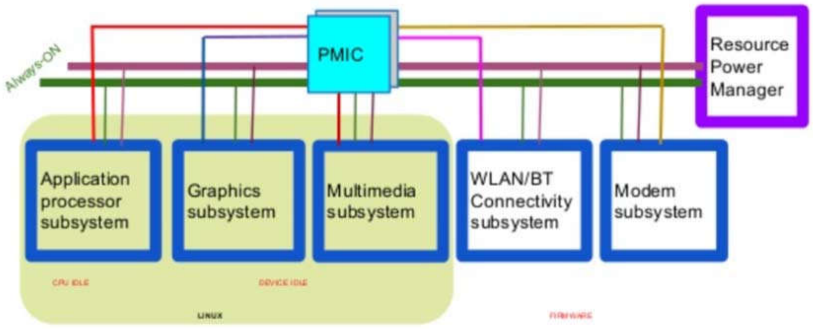
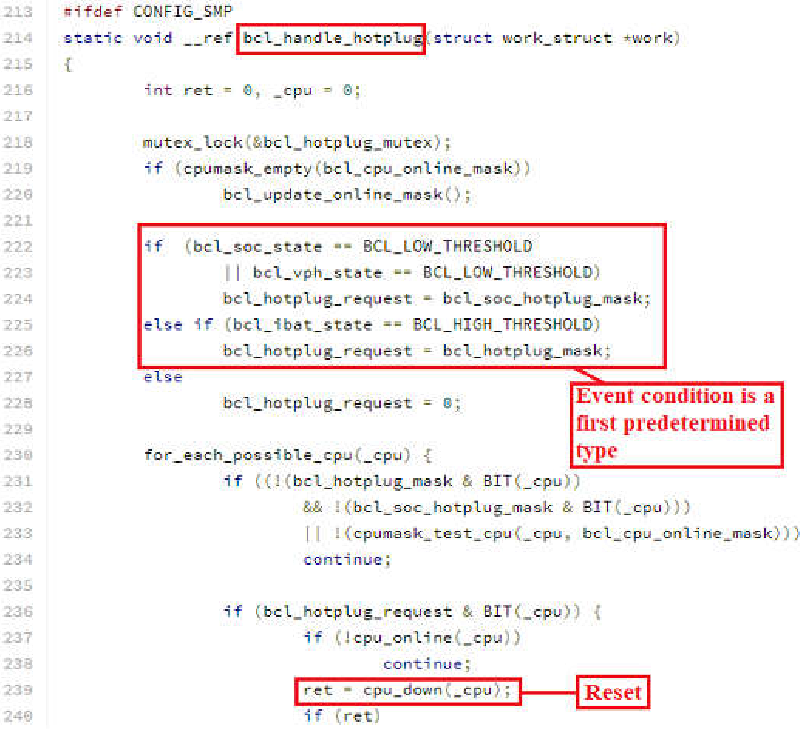
Pat 6,819,539	AT&T's Sale of the LG V30 ("The Accused Product")
	 Resource Power Manager (RPM) RPM is a dedicated hardware engine for managing shared SoC resources, which includes buses, clocks, power rails, etc. The goal of RPM is to achieve the maximum power savings while satisfying the SoC's operational and performance requirements. RPM accepts resource requests from multiple RPM masters. It arbitrates and aggregates the requests, and configures the shared resources. The RPM masters are the application processor, the modem processor, as well as some hardware accelerators. https://android.googlesource.com/kernel/msm/+/android-7.1.0_r0.2/Documentation/arm/msm/rpm.txt  Always-ON PMIC Resource Power Manager Application processor subsystem Graphics subsystem Multimedia subsystem WLAN/BT Connectivity subsystem Modem subsystem CPU IDLE DEVICE IDLE LINUX FIRMWARE https://www.slideshare.net/linaroorg/lcu14-210-qualcomm-snapdragon-power-management-unique-challenges-for-power-frameworks

EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T’s Sale of the LG V30 (“The Accused Product”)</th></tr>
<tr><td></td><td>



```
#ifdef CONFIG_SMP
static void __ref bcl_handle_hotplug(struct work_struct *work)
{
        int ret = 0, _cpu = 0;

        mutex_lock(&bcl_hotplug_mutex);
        if (cpumask_empty(bcl_cpu_online_mask))
                bcl_update_online_mask();

        if  (bcl_soc_state == BCL_LOW_THRESHOLD
                || bcl_vph_state == BCL_LOW_THRESHOLD)
                bcl_hotplug_request = bcl_soc_hotplug_mask;
        else if (bcl_ibat_state == BCL_HIGH_THRESHOLD)
                bcl_hotplug_request = bcl_hotplug_mask;
        else
                bcl_hotplug_request = 0;

        for_each_possible_cpu(_cpu) {
                if ((!(bcl_hotplug_mask & BIT(_cpu))
                        && !(bcl_soc_hotplug_mask & BIT(_cpu)))
                        || !(cpumask_test_cpu(_cpu, bcl_cpu_online_mask)))
                        continue;

                if (bcl_hotplug_request & BIT(_cpu)) {
                        if (!cpu_online(_cpu))
                                continue;
                        ret = cpu_down(_cpu);
                        if (ret)
```

Event condition is a first predetermined type

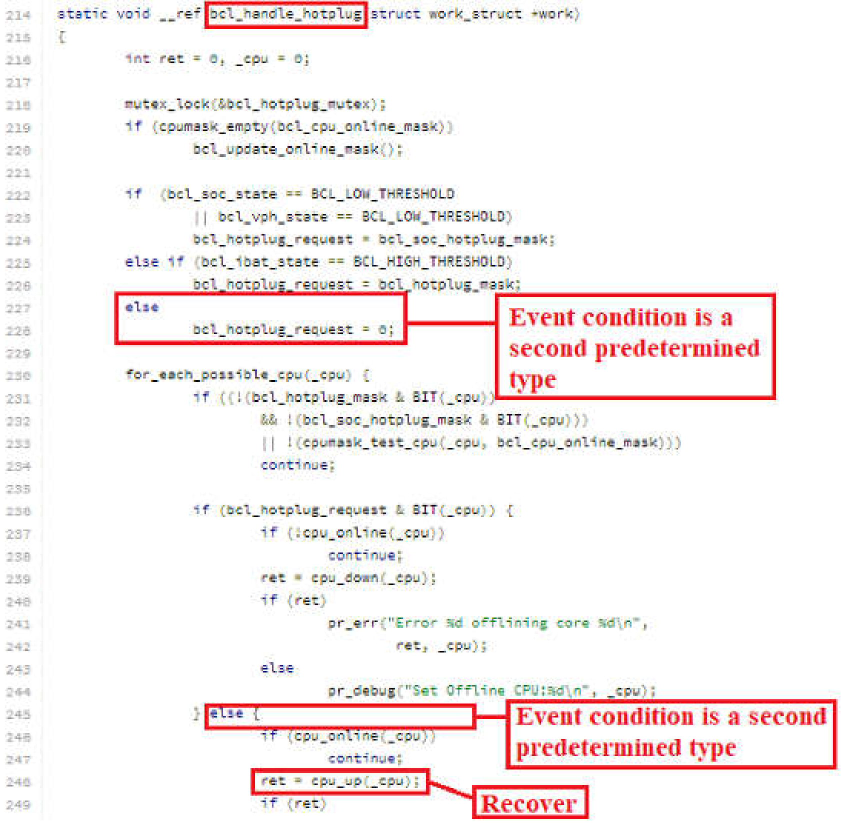
Reset

https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-angler-3.10-nougat/drivers/power/battery_current_limit.c

</td></tr>
</table>

EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T’s Sale of the LG V30 (“The Accused Product”)</th></tr>
<tr><td></td><td>



```
static void __ref bcl_handle_hotplug struct work_struct *work)
{
        int ret = 0, _cpu = 0;

        mutex_lock(&bcl_hotplug_mutex);
        if (cpumask_empty(bcl_cpu_online_mask))
                bcl_update_online_mask();

        if  (bcl_soc_state == BCL_LOW_THRESHOLD
                || bcl_vph_state == BCL_LOW_THRESHOLD)
                bcl_hotplug_request = bcl_soc_hotplug_mask;
        else if (bcl_ibat_state == BCL_HIGH_THRESHOLD)
                bcl_hotplug_request = bcl_hotplug_mask;
        else
                bcl_hotplug_request = 0;

        for_each_possible_cpu(_cpu) {
                if ((!(bcl_hotplug_mask & BIT(_cpu))
                        && !(bcl_soc_hotplug_mask & BIT(_cpu)))
                        || !(cpumask_test_cpu(_cpu, bcl_cpu_online_mask)))
                        continue;

                if (bcl_hotplug_request & BIT(_cpu)) {
                        if (!cpu_online(_cpu))
                                continue;
                        ret = cpu_down(_cpu);
                        if (ret)
                                pr_err("Error %d offlining core %d\n",
                                        ret, _cpu);
                        else
                                pr_debug("Set Offline CPU:%d\n", _cpu);
                } else {
                        if (cpu_online(_cpu))
                                continue;
                        ret = cpu_up(_cpu);
                        if (ret)
```

Event condition is a second predetermined type

Event condition is a second predetermined type

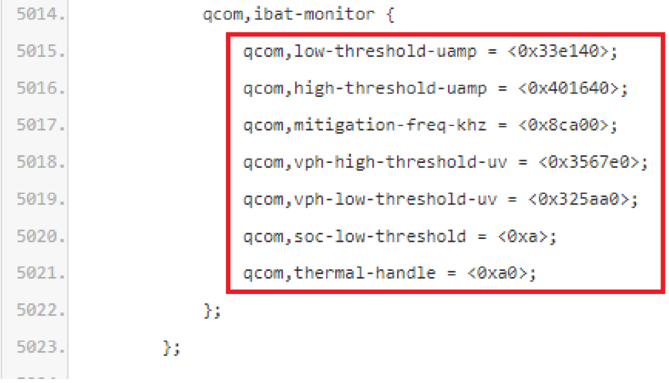
Recover

https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-angler-3.10-nougat/drivers/power/battery_current_limit.c

</td></tr>
</table>

EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T's Sale of the LG V30 ("The Accused Product")</th></tr>
<tr><td></td><td>



```
        qcom,ibat-monitor {
            qcom,low-threshold-uamp = <0x33e140>;
            qcom,high-threshold-uamp = <0x401640>;
            qcom,mitigation-freq-khz = <0x8ca00>;
            qcom,vph-high-threshold-uv = <0x3567e0>;
            qcom,vph-low-threshold-uv = <0x325aa0>;
            qcom,soc-low-threshold = <0xa>;
            qcom,thermal-handle = <0xa0>;
        };
    };
```

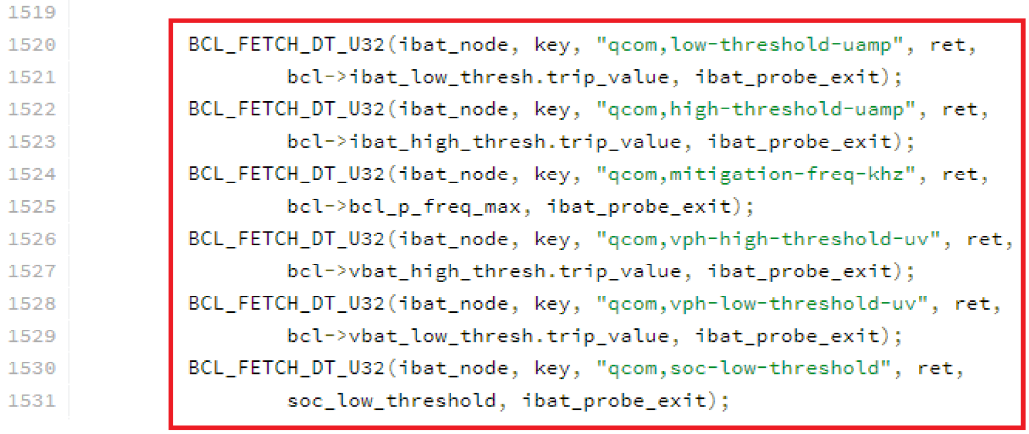
https://pastebin.com/U0i7nP4P

Threshold Values from the table (dtsi) are imported into the battery_current_limit module thru a record data type (bcl).

</td></tr>
</table>

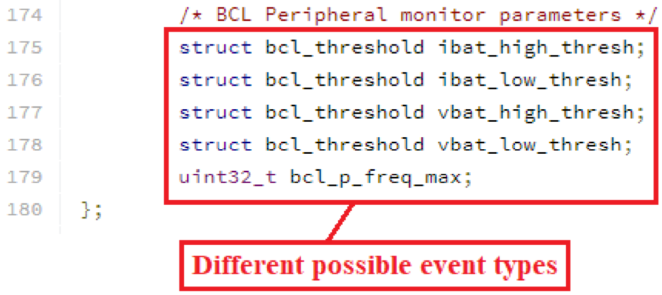
EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T’s Sale of the LG V30 (“The Accused Product”)</th></tr>
<tr><td></td><td>



```
        BCL_FETCH_DT_U32(ibat_node, key, "qcom,low-threshold-uamp", ret,
                bcl->ibat_low_thresh.trip_value, ibat_probe_exit);
        BCL_FETCH_DT_U32(ibat_node, key, "qcom,high-threshold-uamp", ret,
                bcl->ibat_high_thresh.trip_value, ibat_probe_exit);
        BCL_FETCH_DT_U32(ibat_node, key, "qcom,mitigation-freq-khz", ret,
                bcl->bcl_p_freq_max, ibat_probe_exit);
        BCL_FETCH_DT_U32(ibat_node, key, "qcom,vph-high-threshold-uv", ret,
                bcl->vbat_high_thresh.trip_value, ibat_probe_exit);
        BCL_FETCH_DT_U32(ibat_node, key, "qcom,vph-low-threshold-uv", ret,
                bcl->vbat_low_thresh.trip_value, ibat_probe_exit);
        BCL_FETCH_DT_U32(ibat_node, key, "qcom,soc-low-threshold", ret,
                soc_low_threshold, ibat_probe_exit);
```

The values of the table are now inside the record, bcl. The State of Charge low threshold is saved in a variable soc_low_threshold.



```
        /* BCL Peripheral monitor parameters */
        struct bcl_threshold ibat_high_thresh;
        struct bcl_threshold ibat_low_thresh;
        struct bcl_threshold vbat_high_thresh;
        struct bcl_threshold vbat_low_thresh;
        uint32_t bcl_p_freq_max;
};
```

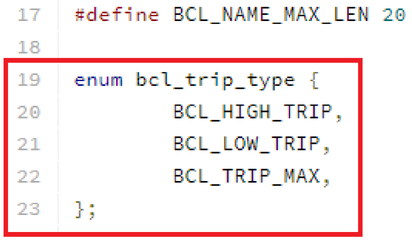
Different possible event types

https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-angler-3.10-nougat/drivers/power/battery_current_limit.c

</td></tr>
</table>

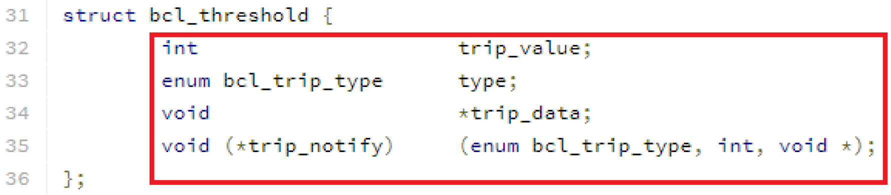
EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T’s Sale of the LG V30 (“The Accused Product”)</th></tr>
<tr><td></td><td>



```
#define BCL_NAME_MAX_LEN 20

enum bcl_trip_type {
        BCL_HIGH_TRIP,
        BCL_LOW_TRIP,
        BCL_TRIP_MAX,
};
```

https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-angler-3.10-nougat/include/linux/msm_bcl.h

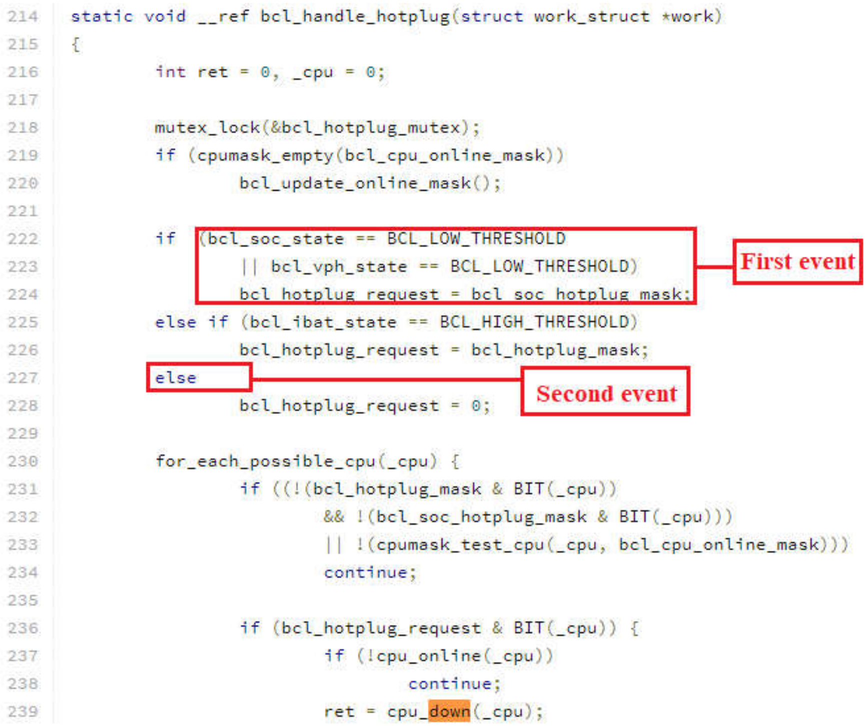


```
struct bcl_threshold {
        int                     trip_value;
        enum bcl_trip_type      type;
        void                    *trip_data;
        void (*trip_notify)     (enum bcl_trip_type, int, void *);
};
```

</td></tr>
</table>

EXHIBIT B

<table>
<tr><th>Pat 6,819,539</th><th>AT&T's Sale of the LG V30 ("The Accused Product")</th></tr>
<tr><td></td><td>



```
static void __ref bcl_handle_hotplug(struct work_struct *work)
{
        int ret = 0, _cpu = 0;

        mutex_lock(&bcl_hotplug_mutex);
        if (cpumask_empty(bcl_cpu_online_mask))
                bcl_update_online_mask();

        if (bcl_soc_state == BCL_LOW_THRESHOLD
                || bcl_vph_state == BCL_LOW_THRESHOLD)
                bcl_hotplug_request = bcl_soc_hotplug_mask;
        else if (bcl_ibat_state == BCL_HIGH_THRESHOLD)
                bcl_hotplug_request = bcl_hotplug_mask;
        else
                bcl_hotplug_request = 0;

        for_each_possible_cpu(_cpu) {
                if ((!(bcl_hotplug_mask & BIT(_cpu))
                        && !(bcl_soc_hotplug_mask & BIT(_cpu)))
                        || !(cpumask_test_cpu(_cpu, bcl_cpu_online_mask)))
                        continue;

                if (bcl_hotplug_request & BIT(_cpu)) {
                        if (!cpu_online(_cpu))
                                continue;
                        ret = cpu_down(_cpu);
```

https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-angler-3.10-nougat/drivers/power/battery_current_limit.c

The new values of bcl_vph_state and bcl_ibat_state are compared against the threshold values from the table.

</td></tr>
</table>

EXHIBIT B